SUN BLUEPRINTS APPROVED

Sun ONE Portal Server 6 Best Practices

Christian Candia—Sun Professional Services

Sun BluePrints™ OnLine—October 2003

`http://www.sun.com/blueprints`

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
(650) 960-1300

Part No. 817-3836-10
Revision 06, 10/8/03
Edition: October 2003

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Please
Recycle

Adobe PostScript™

Sun ONE Portal Server 6 Best Practices

When proposing a technical solution for an specific problem, the first step is to collect functional and nonfunctional requirements. Generally, these requirements fall into the following categories:

- Performance
- Risk
- Cost
- Schedule

Most of the time, especially in complex systems, such as portals where content is aggregated from many different sources, these requirements conflict with each other. Usually security and ease of use call for different approaches. A very secure site can be hard to use because it requires complicated, hard-to-remember passwords, or there are stringent session and inactivity timeouts. In addition, availability and performance sometimes conflict. For example, to provide session failover, it is necessary to keep the session in sync on all of the servers. This synchronization adds some delay to each transaction.

The process of creating a solution involves understanding the trade-offs between all conflicting requirements and deciding what is more important for a successful implementation. This article presents some architectural guidelines that are frequently applied to Sun™ ONE Portal Server 6 software implementations and will help you to identify and understand potentially conflicting requirements on the performance and risk categories. After you understand these categories, you should be able to include the cost and schedule requirements when you define the final solution.

This article presents the best practices for high availability, security, and scalability that commonly have more impact on the success of a Sun ONE Portal Server software solution. In addition, the article includes guidelines for creating a Sun ONE Portal Server software solution that can be easily supported.

Before you read this article, you should have a detailed technical understanding of the Sun ONE Portal Server 6 software and the Sun ONE Portal Server Secure Remote Access components, such as the Gateway, the Netlet, the Rewriter proxies, and the search engine. Also, an in-depth knowledge of the embedded Sun ONE software products (for instance, the Sun™ ONE Directory Server, the Sun™ ONE Web Server, and the Sun™ ONE Identity Server) is required.

High Availability

Delivering high services levels is a top priority for all Sun ONE Portal Server software implementations. You can determine the availability of a system by using a simple equation, as shown in FIGURE 1.

$$\text{Availability} = \left[\frac{\text{Uptime}}{(\text{Uptime} + \text{Downtime})}\right] * 100$$

FIGURE 1 Availability Equation

As the equation shows, if you decrease the downtime of the system, you can increase the availability of the system. However, when you measure the downtime, you must measure the total amount of time the system is unavailable, which should include the planned downtime (for example, maintenance, backups, and repairs to the system) and the unplanned downtime (for example, system or network failures). Some studies show that planned downtime can account for up to 80 percent of the total time a system is unavailable.

Thus, when you are architecting a solution, you must consider both the planned and unplanned downtime to ensure that you create a highly available solution. Availability is affected by all of the components in the system, such as the following infrastructure components:

- Hardware
- Network
- Operating system
- Applications

In addition to these infrastructure components, availability is also affected by people and processes, so when you are architecting a highly available solution, you must ensure that the people who will be supporting the solution have the proper training and skill sets, and you must ensure that clearly defined processes are in place to support the system.

For background information on the concept of availability, refer to “Availability - What It Means, Why It’s Important, and How to Improve It” (Sun BluePrints™ OnLine, 1999).

In reference to availability, system types can be defined in four ways: noncritical, task critical, business critical, and mission critical. The noncritical system type is a basic system that has no requirements for availability. If the system goes down, it can be repaired in a matter of days without affecting users. This type of system is not important to the discussion of availability in this article.

Task-Critical Systems

Unlike the noncritical system, the task-critical system does have availability requirements. If the system goes down, it would affect users, and the performance of the system could be affected. The best way to achieve the availability levels required for this kind of system is by using redundancy of services. To optimize the usage of the system resources, all of the redundant components should be active (that is, they should not be in standby mode). Replication, load balancing, and service redundancy must be used to achieve this goal. FIGURE 2 shows the basic design of a task-critical system.

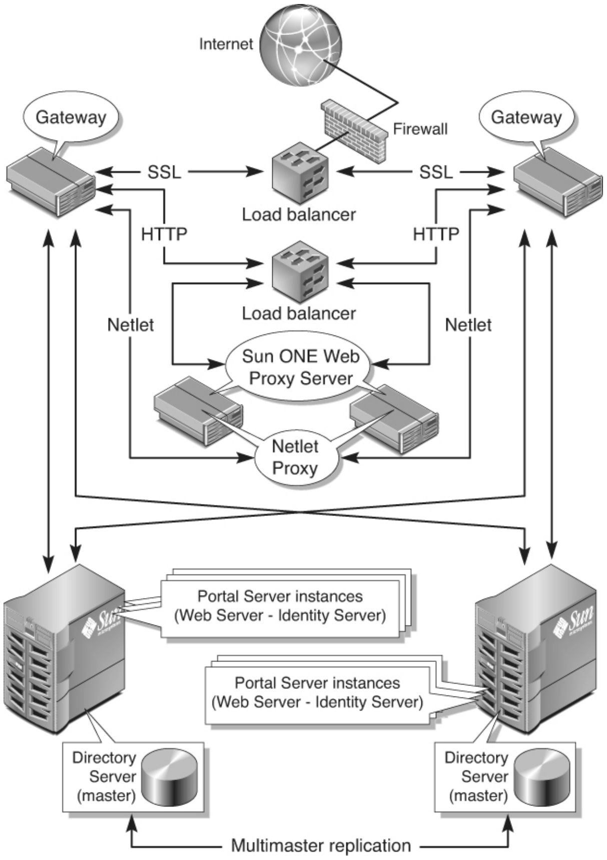
Internet
Gateway
Firewall
Gateway
SSL
SSL
Load balancer
HTTP
HTTP
Load balancer
Netlet
Netlet
Sun ONE Web Proxy Server
Netlet Proxy
Portal Server instances (Web Server - Identity Server)
Portal Server instances (Web Server - Identity Server)
Directory Server (master)
Directory Server (master)
Multimaster replication

FIGURE 2 Task-Critical System Diagram

Gateway Server Availability

As FIGURE 2 shows, in this architecture, there are at least two gateway servers that are *front-ended* by a load balancer so that all of the requests are spread across the gateways. The load balancer must also be configured to detect failures in the gateways. If a gateway fails, then the load balancer sends all of the requests to the surviving gateway.

The gateways are a stateless process, so if a gateway fails, all of the sessions associated with that gateway can be redirected to the other gateway. Users will not perceive any downtime because the Portal Server session is maintained on the Portal Server nodes, not in the gateways.

When you use gateways, it is likely that the resource servers that are being accessed through the gateways will be on a private network that is protected by a firewall. In this case, you might want to use a web proxy to access these resource servers so that a single hole is open in the firewall. Even though the Sun ONE Portal Server software includes a rewriter proxy, it is not a fully functional web proxy server. For example, it does not support caching, the Internet Caching Protocol (ICP), and URL filtering. However, the Sun ONE Web Proxy Server software is a reliable, inexpensive, and highly configurable web proxy server, which in addition to these features, provides generic protocol support for a firewall traversal by using SOCKSv5.

Portal Server Instances

In the architecture depicted in FIGURE 2, the Portal Server instances are installed on the Sun ONE Web Server as web containers. The Sun ONE Web Server software does not support replication of user sessions across instances, so when a Portal Server instance goes down, all of the sessions maintained on that instance are lost. The same is true if the web container used is an application server, such as the Sun ONE Application Server Standard Edition, that does not support session failover.

To increase availability of the Portal Server, you can create multiple instances of the Sun ONE Web Server on the same machine or have multiple instances on multiple machines. In this way, the number of users affected by a Portal Server instance failure is minimized. Users that are affected would have to log in to another server.

If an instance fails, the gateway detects the failure and reroutes the requests to one of the surviving instances. If you are not using the Sun ONE Portal Server Secure Remote Access software, you must have load balancers to perform the functions of the gateways, and the load balancers have to detect the failure of a Portal Server instance and send the requests to one of the surviving instances.

Directory Server Availability

Another important component of the Sun ONE Portal Server software solution is the Directory Server where the user and services profiles are stored. To remove this component as a single point of failure, you can use the Sun ONE Directory Server software’s multi-master replication (MMR) configuration or the Sun™ Cluster software framework. Because of the loosely consistent replication mechanism of the Sun ONE Directory Server software, it is possible, albeit very unlikely, that an update can be lost. If a system failure occurs right after a change has been accepted by one master, but before the change is replicated to the second master, it is possible that the change will be lost, and there is no easy way to detect this fact.

In some very demanding environments, the possibility of losing an update might not be acceptable. In this case, the best option is to use the Sun Cluster software to achieve high availability of the Directory Server. The use of the Sun Cluster software can increase the availability of the system, but the configuration, maintenance, and monitoring of this environment require more specialized knowledge and very well defined operational processes.

When you are installing the Portal Server software, you can only specify one LDAP server, and this server must be a master LDAP server because the installation process is affected by the propagation delay of the LDAP replication process. To add multiple LDAP masters or to point the Sun ONE Identity Server to use a consumer after the installation, you must edit the `serverconfig.xml` file and add a *Server* element for each additional LDAP server. The following example shows the format of the server entries:

```
<Server name=Server1 host=master1.company.com port=389 type=SIMPLE />
<Server name=Server2 host=master2.company.com port=389 type=SIMPLE />
```

The Identity Server uses the first entry as the LDAP server for all requests of service, roles, organization, and user profiles. If that LDAP server fails, the Identity Server fails over to the next server in the list. There is no round-robin or failback between the LDAP servers, so if you want to design a solution in which all of the LDAP servers are used evenly, you will have to use the Sun ONE Directory Proxy Server software. A load balancer cannot be used because the Sun ONE Identity Server software uses a pool of connections that are kept open and are reused. The same is true for the LDAP and membership authentication modules and for the Policy Configuration service. They can use primary and backup LDAP servers, but you have to add the failover servers after the installation by using the administration console.

Planned Downtime

With the architecture shown in FIGURE 2 on page 4, there is redundancy of services, so most of the unplanned downtime can be minimized or eliminated. However, the planned downtime is still an issue. For instance, if the Portal Server software must be updated, services could be affected. If the upgrade or patch includes changes to the Sun ONE Directory Server software schema used by the Sun ONE Identity Server software, all of the software components must be stopped to update the information stored in the Directory Server.

In addition, the Solaris™ Operating System (Solaris OS) patch installation process does not work if the application services are enabled. Thus, you must shut down all of the services, patch the system, then bring the system back online. In some environments, the downtime incurred during the patch process might not be acceptable. But, with a highly available solution with duplicate services and components, you can use a phased approach for maintaining the system. For instance, you could remove one Portal Server node from the production configuration and upgrade it. Then, you could remove one of the gateways from the production configuration and upgrade it. Afterwards, you could integrate that silo back into the production configuration and repeat the process for the other silo, resulting in an upgrade with minimal interruption of service.

In theory, you would not have downtime; however, because of the architecture of the Portal Server software, it is not possible to just remove one Portal Server instance from the active configuration without affecting some users because there is no way to prevent a user from logging in to the server and to keep the active sessions untouched. Thus, you must create a mechanism to prevent users from logging in to the server while the gateways still process request from the already-authenticated users. This can be accomplished by using a custom-developed authentication module.

Business-Critical Systems

The third type of system is the business-critical system. For this type of system, availability is a critical requirement because if the system goes down, it could lead to lost revenue, lost productivity, and customer dissatisfaction. FIGURE 3 shows the typical configuration of a business-critical system, which builds on the architecture described for the task-critical system and includes all of its benefits.

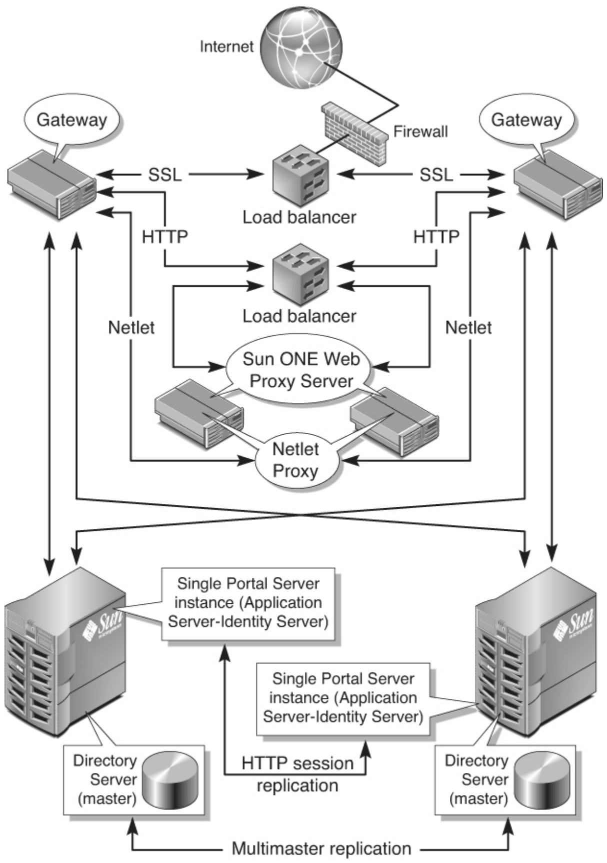
Internet
Firewall
Gateway
Gateway
SSL
SSL
Load balancer
HTTP
HTTP
Load balancer
Netlet
Netlet
Sun ONE Web Proxy Server
Netlet Proxy
Single Portal Server instance (Application Server-Identity Server)
Single Portal Server instance (Application Server-Identity Server)
Directory Server (master)
HTTP session replication
Directory Server (master)
Multimaster replication

FIGURE 3 Business-Critical System Configuration

To enhance the availability of the system, an application server is used as a web container for all of the Identity Server and Portal Server services. This configuration is needed to use the HTTP session failover features of the Application Server. This maintains a database of all of the sessions that are created in the system, and that database is accessible to all of the Portal Server instances in the configuration. Thus, if one Portal Server instance fails, the gateways redirect all of the requests to the surviving Portal Server, and that Portal Server will be able to validate that the session is still valid so that the operation will continue to work smoothly.

In the architecture depicted in FIGURE 3, the availability of the system is much higher than the architecture discussed in the “Task-Critical Systems” section. However, it is not possible to achieve absolute session failover. Depending on how an application server instance fails and how and where user sessions are stored and replicated, newly created individual user sessions might not have been written to the common database, so they might be lost.

The Sun ONE Portal Server software, version 6.2, supports BEA's WebLogic 6.1 and the Sun™ ONE Application Server 7.0 Enterprise Edition for session failover. When using BEA's application server, the WebLogic Cluster software is required to create an environment in which the sessions are replicated; however, the Sun ONE Portal Server Secure Remote Access software does not work with the WebLogic Cluster software. Thus, it is not possible to implement a highly available solution with the WebLogic software if the gateways are also required.

For the business-critical system, upgrades have a minimum impact. Either Portal Server node can be taken out of the production configuration without affecting the users because the sessions are in the shared database and because requests will be handled by the available Portal Server node.

Mission-Critical Systems

The fourth type of system is the mission-critical system. For the mission-critical system, failures could have catastrophic results for an organization (for example, loss of life or serious injury, significant loss of money, serious inability to conduct business, or serious operational chaos). Most mission-critical systems are usually custom built using special hardware such as fault-tolerant computers and software.

Security

Data loss, electronic snooping, hacker attacks, unauthorized access, stolen passwords, and denial of service are just a few of the security issues a portal solution can face. Protecting the integrity and confidentiality of information is critical. To do this, the networking environment in which the portal resides must be secure. This section includes descriptions of how security should affect the configuration of each component in a Portal Server system.

Security is not just about the infrastructure of a system, it is also about the people, processes, policies, and architectures. System administrators must be trained in security-related issues. The processes must account for security issues, and the policies must include directives that will help to prevent security attacks. Finally, the architectures must be secured against both external and internal attacks.

Hardening and Minimization

Minimization is especially important in an environment that is exposed to the Internet or any untrusted network. This is the case on most implementations of the Sun ONE Portal Server software. In these environments, it is very important to reduce the Solaris OS installation to the minimum number of packages necessary to support the hosted applications. This minimization in services, libraries, and applications helps increase security by reducing the number of subsystems that must be disabled, patched, and maintained.

Solaris Security Toolkit

The Solaris™ Security Toolkit software, also known as JASS, is a tool designed to assist in the development, deployment, and maintenance of a secured Solaris OS. The Toolkit includes a set of shell scripts that implement the recommendations that are outlined in “Solaris Operating Environment Network Settings for Security: Updated for Solaris 9 Operating Environment” (Sun BluePrints OnLine, June 2003).

For detailed information about the Toolkit, refer to *Securing Systems with the Solaris Security Toolkit* (Prentice Hall, 2003). Used in conjunction with a JumpStart™ software server, the Toolkit can be used to install, minimize, and harden a Solaris OS server. The Toolkit is used during the OS installation process by using JumpStart software finish scripts. These scripts are executed after all of the software packages are installed.

Required Solaris OS Packages

The Sun ONE Portal Server software framework requires only an small subset of the Solaris OS packages to work properly. For Sun ONE Portal Server software, version 6.1, in addition to packages bundled in the Solaris OS Core software group (`SUNWreq`), only `SUNWlibC`, `SUNWadmc`, and `SUNWadmfw` are required to support most of the Sun ONE Portal Server software components. In addition, if the web container used by the Sun ONE Portal Server software, version 6.1, is the Sun ONE Application Server 7 software, a number of extra packages are required (see TABLE 1).

TABLE 1 Required Packages for Use With the Sun ONE Application Server 7 Software

Package	Description
`SUNWpl5u`	Perl 5.6.1 (core)
`SUNWuiu8`	Iconv modules for UTF-8 Locale
`SUNWuiu8x`	Iconv modules for UTF-8 Locale (64-bit)

The Sun ONE Portal Server software, version 6.2, is part of the Java™ Enterprise System, which uses a common installer for all Sun ONE software applications. To install version 6.2 of the Sun ONE Portal Server, the packages in TABLE 2 are required.

TABLE 2 Required Packages for Use With the Sun ONE Portal Server 6.2 Software

Package	Description
`SUNWadmc`	System administration (core)
`SUNWadmfw`	System and Network Administration Framework
`SUNWlibC`	Sun Workshop Compilers bundled (`libC`)
`SUNWlibCx`	Sun Workshop Compilers bundled (64-bit, `libC`)
`SUNWgzip`	The GNU Zip (`gzip`) compression utility
`SUNWfns`	Federated naming system
`SUNWfnsx`	Federated naming system (64-bit)
`SUNWgss`	GSSAPI V2
`SUNWgssx`	GSSAPI V2 (64-bit)
`SUNWzlib`	The Zip compression library
`SUNWzlibx`	The Zip compression library (64-bit)
`SUNWscpu`	Source compatibility (`usr`)

Netmail is a Java™ technology-based applet that implements a GUI that serves as a front end to an IMAP4-compliant mail server. The applet uses a servlet to interface with the mail server. This servlet is installed on the same web container that supports the Sun ONE Portal Server and Sun ONE Identity Server software. The servlet uses JNI and the AWT graphic toolkit that is part of the OpenWindows™ software and Motif windowing environments. Consequently, the Solaris OS packages that contain the AWT libraries and its dependencies also must be installed on the system if Netmail is used. TABLE 3 lists the required Solaris 9 OS packages.

TABLE 3 Required Packages for Netmail

Package	Description
`SUNWxwpllt`	X Window system platform software
`SUNWxwice`	X Window system Inter-Client Exchange (ICE) components
`SUNWmfrun`	Motif runtime kit

Netfile is a Java technology-based file manager application that enables users to have remote access to FTP, SMB, and NFS based file servers. For Sun ONE Portal Server 6.1 software, to access Microsoft Windows file servers, Netfile uses the `smbclient` program that is included in the SAMBA open source software suite. For convenience, a version of SAMBA is included in the Sun ONE Portal Server software third-party CD in the SUNWsmbac package. Version 6.2 of the Sun ONE Portal Server software uses the jCIFS toolkit of the SAMBA suite, so there is no need for this additional package.

To access NFS servers, the Solaris OS packages that contain the NFS client application and libraries must be installed on the system. TABLE 4 contains the list of required packages.

TABLE 4 Required Packages for NFS Support

Package	Description
SUNWnfscr	NFS client support (root)
SUNWnfscu	NFS client support (usr)
SUNWnfscx	NFS client support for 64-bit systems (root)

Application Ownership

By default, the installation of the Sun ONE Portal Server software is done as the system superuser (root). All of the components of the Sun ONE Portal Server are installed and configured to run as root. Unfortunately, there are some security implications for having the processes running as root. An application bug can be exploited to gain root access to the system. root access is required to maintain the Sun ONE Portal Sever software application. This raises potential security concerns because this responsibility is typically delegated to non-system administrators who might pose a threat to the system security and integrity.

The Sun ONE Portal Server software documentation contains instructions on how to change the user ownership of the Sun ONE Portal Server software processes and files. The document assumes that the same UNIX® user will run all of the services. In general, it is not recommended to run any application service as the nobody user. A better approach is to run the different applications under different users. This ensures that the maintenance of the different components will be done by different groups of administrators. Instead of creating dedicated or *functional* users for each application, it is better to assign the applications to different UNIX roles so that users can be assigned and removed from the role as needed.

While the traditional UNIX security model is generally viewed as *all or nothing*, there are tools that can be used as an alternative to provide additional flexibility. These tools provide the mechanisms needed to create a fine-grained access control system in which users can be selectively granted access to individual resources, such as different UNIX commands.

The Solaris 8 OS and the Solaris 9 OS support Role-Based Access Control (RBAC), which provides the ability to package superuser privileges and assign them to user accounts. RBAC enables separation of powers, controlled delegation of privileged operations to other users, and a variable degree of access control. This feature of RBAC enables you to configure the system using default ports (389 for the Directory Server and 80 for the Portal Server) and have a non-superuser role dedicated to the management of the application.

Secure Shell

As part of the Solaris Security Toolkit software minimization process, the common network access protocols, such as Telnet and FTP, are disabled because there is no way to prevent passwords and data from being transmitted in clear text. Thus, these protocols are susceptible to eavesdropping. The Toolkit also disables the `rlogin`, `rcp`, and `rsh` commands. The recommended tool used to provide remote access to a server is Secure Shell.

Secure Shell encrypts all network traffic, provides strong authentication, and monitors the integrity of the network session. It provides equivalent replacements for common commands such as `telnet`, `ftp`, and `rcp`. A description of the features and different configuration options for Secure Shell can be found in *Secure Shell in the Enterprise* (Prentice Hall, 2003). The Solaris™ Secure Shell software is a bundled component of the Solaris 9 OS, and it is part of the companion CD in the Solaris 8 OS.

Secure Socket Layer (SSL)

To increase the security of the Portal Server, HTTP over SSL (HTTPS) can be enabled on the Portal Server nodes. The Sun ONE Web Server 6 software supports multiple listening sockets that can be associated with the same virtual server. These listening sockets can be configured to use either HTTP or HTTPS protocols. Thus, it is possible to configure the Portal Server to user only HTTP, only HTTPS, or both at the same time on different ports.

The best option to create a portal site that will use SSL is to do it during the software installation. The installer will configure all of the Sun ONE Portal Server and Sun ONE Identity Service software services to use HTTPS. After the software is installed, everything is configured to use HTTPS, but the X.509 certificate for the server and the security for the listening socket on the Web Server must be manually enabled using the administration console on the Web Server.

LDAP over SSL (LDAPS) can also be used to guarantee integrity and confidentiality in the access to the information stored in the Directory Server. The Sun ONE Directory Server software can be configured to listen on both LDAP and LDAPS default ports simultaneously (ports 389 and 636, respectively).

Setting the LDAP port to 0 completely disables the non-SSL access. This is the recommended configuration because it guarantees that all LDAP requests will be protected by the SSL protocol. In this configuration, the Directory Server administration server also must be configured to use LDAPS to access the directory. At the same time, access to the administration server should be done using HTTPS.

SSL version 2 and 40-bit and 56-bit ciphers are disabled by default because of known deficiencies in the protocol implementation and the poor security that those weak ciphers provide. Most browsers use RC4 as the default cipher for encryption because it is the fastest cipher and because at the time SSL version 3 was published, RC4 was fairly secure. However, since its release, some vulnerabilities have been discovered that might make it theoretically possible to recover the encryption key from the encrypted data. The 3DES cipher is slower, but it is more secure because it has no known vulnerabilities. You will need to take the customer’s need for performance versus the customer’s need for security into account when you decide which cipher to enable or disable.

Firewalls

Usually, firewalls are configured to drop connections that have been idle for some predefined time, which varies from minutes to hours. This can cause a problem for Netlet connections if a firewall is between the user’s browser and the gateway. To avoid this problem, the Netlet keep-alive attribute must be set to a time shorter than the firewall timeout. This will force the gateway to send a package to the Netlet applet that will reset the idle timer of the firewall.

The same problem can occur if there is a firewall between the Sun ONE Portal Server software node and the server that hosts the Directory Server. The Sun ONE Identity Server software uses a pool of open connections to access the LDAP server. If any connection is idle for a longer time than the idle timeout of the firewall, the connection will be closed. The Identity Server will believe the LDAP server is down and will failover to the next LDAP server if it is configured. To prevent this problem, you should set the maximum connection time in the LDAP server.

In previous versions of the Sun ONE Portal Server software (prior to version 6.2), the Netlet used a proprietary protocol that mimics the SSL semantics, but did not implement the SSL handshake protocol, which is required for some firewalls to keep track of the valid SSL sessions. Because of this reason, the Netlet component did not work with proxy firewalls. These firewalls analyze every protocol encapsulated in the network packet, and because the Netlet packet did not have a valid SSL session

ID, a proxy firewall would drop the packets. This limitation has been removed in the latest version of the Sun ONE Portal Server Secure Remote Access software. In this version, the Netlet uses the SSL protocol as a transport.

Identity Server Administration Console

The administration console in the Sun ONE Identity Server software is used to configure every aspect of the Identity Server software. It is used to create organization, users, and policies and to modify services. In most cases, it is not desirable that ordinary users have access to the administration console. The best option is to completely disable the administration console on the servers that will be accessed from the Internet and to install it on a dedicated server on a protected network. In version 6.1 of the Sun ONE Portal Server software, there is no option during the installation to not install the administration console. To disable the administrative console, it was necessary to remove the administration console application (amconsole) from the web container. In the Sun ONE Web Server 6 software, this can be done using the following command:

```
# /opt/SUNWam/servers/bin/https/httpadmin/bin/wdeploy delete u /amconsole\
-i https-<Server FQDN> -v https-<Server FQDN> soft
```

The delegated administration and self-service features of the Sun ONE Identity Server software are also accessed through the administration console URL. Removing the administration console will also remove these features from the system.

Penetration Testing

It is common for companies to perform penetration testing to verify that the system complies with all of the security policies and that there are no exploitable security holes. Penetration testing involves intentional hacking into a system to find failures and/or misconfigurations that enable users to open a valid session on the Portal Server or to gain illegal access to the system. This testing must be done by someone who is very familiar with hacking tools and is able to analyze the output of the tools. Even though most, if not all, of the tools are available on the Internet, the planning, execution, and result analysis must be done by a certified UNIX security consultant.

Intrusion Detection

Despite the use of firewalls, a second level of defense is needed. Firewalls are usually configured to allow common Internet traffic (for example, SSL, HTTP, and SMTP). Network attacks can still be made using one of these protocols. For example, HTTP is being used as a transport not only for HTML, but also for SOAP, SAML, and other protocols that can be used to interact with applications. These protocols can be exploited to carry some specific attacks to try to exploit software vulnerabilities.

Intrusion detection involves detecting network security attacks within the system. Specific tools are used to detect the intrusions, and after an intrusion is detected, additional tools are needed to conduct intrusion forensics, nonrepudiation, and selective logging. None of these tools are provided with the Portal Server software. When any of these attacks occur, the intrusion detection system will notice them and will stop the attackers before they can reach systems and data.

In maintaining a secure system, it is important to monitor the system to ensure that files have not been changed. You can use Tripwire to check every file on the system. Tripwire (`http://www.tripwire.com`) can be used to monitor file changes, verify integrity of data, and notify the system administrator of any violations. Tripwire makes it possible to establish network policies that detect intentional tampering, user error, software failure, malicious software, and open-door systems.

In addition, the Solaris OS Software Fingerprint Database (`sfpDB`) is a free SunSolve Online^{SM} service that enables users to verify the integrity of files distributed with the Solaris OS. The checksums of the system files must be updated after the system is modified by patch or software installations. The issue with existing tools has always been verifying that the files used to generate the baseline checksums are correct and current.

The Fingerprint Database addresses the issue of validating the base files provided by Sun. This includes files distributed with the Solaris OS media kits, unbundled software, and patches. The Fingerprint Database provides a mechanism to verify that a true file in an official binary distribution is being used and not an altered version that compromises system security and causes other problems.

Scalability

Scalability is described as how well an architecture will perform when the size of the system increases. Scalability is usually divided in two types: vertical and horizontal. Vertical scaling basically means putting more resources into the system to increase performance. Horizontal scaling basically means adding more servers to the configuration.

For solutions based on the Sun ONE Portal Server software, scalability is one of the factors that will define an architecture. This section describes how each component of the Sun ONE Portal server scales and what the common problems and limitations to overcome are.

Portal Server Instance Scalability

The Sun ONE Portal Server software can scale up to four CPUs for a single instance. This limit is due to the garbage collection system of the underlying Java VM, which becomes a bottleneck under heavy loads. If the Sun ONE Web Server is used as a web container, the Sun ONE Portal Server software can support multiple instances of the web server on the same machine. In this way, it can it can scale vertically.

When an Application Server is used as a web container, multiple instances on the same machine are not supported, so the only way to scale is horizontally by adding more servers to the configuration. In both cases, a load balancer or gateway servers should be used to provide a *single system* image.

The Sun ONE Portal Server software distribution contains a tuning utility that automates the tuning of a Portal Server node. The `perftune` script tunes the Solaris OS kernel and TCP parameters, and it modifies the Sun ONE Web Server software, the Sun ONE Directory Server software, the Sun ONE Identity Server software, and the Sun ONE Portal Server software configuration files.

The script implements two tuning strategies:

- Production optimum for a high level of user requests from a small number of users
- Production large for a low level of requests from a large number of users

The perftune script changes the application's configuration parameters to values that had been found to generally increase portal throughput, but it is possible that further tuning will be required to achieve optimum performance. These changes must be tested and validated on each portal installation. Such tuning should be done by someone who has an in-depth understanding of the Portal Server component products.

Gateway Scalability

The gateways can scale both vertically by running multiple instances of the Java technology-based process that implements the gateway and horizontally by adding more servers to the configuration. The best option is to use multiple small servers with up to four CPUs for the gateways, accessed through a load balancer. In this way, not only is it easy to scale the system if more power is required by adding another server to the configuration, but the load balancer can also failover requests if the gateway server goes down. The same applies to the Netlet and Rewriter proxies, except that a load balancer is not required because the gateways will distribute the connections in a round-robin manner among all of the available Netlet and Rewriter proxies.

The only case in which it is recommended to use multiple gateway instances on the same server is when the gateway must be accessed through different URLs. In this case, each URL can have its own X.509 certificate associated with it. The only way to implement this is with multiple gateway instances because each instance uses a single certificate.

The gateways have been tested with most commercial load balancers, such as Alteon, Cisco CSS, BigIP, and Central Dipatch. However, the load-balancing requirement for the gateway is very basic: load balancing should be based on the SSL session ID. There is no need for the load balancer to analyze the HTTP header or use complex content management operations. The most cost-effective solution is to use an inexpensive load balancer that supports persistence based on the SSL session ID. The use of a software load balancer is probably the best option because it can be installed on the same server that is supporting the gateway; thus, there is no requirement to purchase, monitor, or maintain additional hardware.

Directory Server Scalability

To avoid a bottleneck in the access to the Directory Server, especially when multiple Portal Server instances are used, you should use a Directory Server instance installed on each Portal Server node. Without it, the requests from the Identity Server to the LDAP server could saturate the network connection between the Portal Server and the Directory Server. If a consumer is used, the replication protocol will be more efficient than pointing the Identity Server to use an LDAP server on a different machine. The only caveat is that the Identity Server administration console must be used against an LDAP master server because the Identity Server is affected by the propagation delay of the replication protocol.

SSL Accelerators

SSL encryption and decryption can use a lot of processor cycles, limiting the number of SSL transactions per second that the system can sustain. To increase the ability to handle SSL transactions, as well as to reduce processor overhead, it is common to offload SSL processing from the server to a dedicated board installed on the system (internal SSL accelerator) or to a network device (external SSL accelerator).

External SSL accelerators, such as Cisco's CSS SSL module, are not supported on the Portal Server instances. Internal SSL accelerators, such as the Sun™ Crypto Accelerator 1000 board, will only accelerate the establishment of an SSL session, not the bulk encryption. Thus, SSL accelerators are a good option for portals for which there are a lot of short-lived sessions (that is, sessions lasting only a few minutes). If sessions are kept open for long periods of time, the accelerators will provide very little relief.

The Netlet component of the Sun ONE Portal Server software also does not support external accelerators, but you can use internal SSL accelerators. However, because the Netlet is used to proxy TCP sessions that tend to be of long duration, it is very unlikely that an SSL accelerator will provide significant gains in performance.

Sizing

The sizing of a Portal Server is an extremely difficult task because it is impossible to test and benchmark all of the applications that can be sent through the portal solution. Sizing is also difficult because each customer might want to integrate applications in different ways, possibly using their own customizations. There is a Sun internal Portal Server sizing tool, as mentioned in the *Sun ONE Portal Server Deployment Guide*. This sizing tool was built using benchmarks that simulated a limited number of possible scenarios.

To adapt the results from the tools to the reality of a given Portal Server, the sizing tool requires you to use a *SHARP factor* to propose the number of CPUs that are needed to support the Portal Server software. The SHARP factor summarizes in a single number the differences between your environment and the environment used as the baseline for factoring the sizing tool. Because the performance details of the desired production environment are not known and the scenarios used to build the sizing tool are not fully disclosed, a certain amount of guesswork is needed to obtain a final configuration using this tool.

A more scientific approach is to use the concept of building blocks, as explained in the *Sun ONE Portal Server Deployment Guide*. You can use the sizing tool to determine a building block size (that is, what kind of servers should be used as the Portal Server nodes). Then, you can use that building block size to implement a pilot that integrates most of the applications that will be accessed through the Portal Server and test this configuration with a limited number of real users.

The usage patterns of the users can be collected and used to create a load test script for any commercial load-testing tool. The test script enables you to obtain a detailed load curve for the building block. With this information, you can size your production architecture, based on the applications to be integrated and the expected number of users of the system. You can also use the load curve to predict when a building block needs to be added to the production architecture or when the number of users reaches a certain limit.

Supportability

Supportability of the Sun ONE Portal Server software is critical to achieve the required performance levels of the proposed solution. To achieve this, you must consider the establishment of a preproduction environment, the creation of organization-specific directories to contain the templates and pages that are used to implement the look and feel of the organization, the use of a well-defined process to install customizations on every production server, and the creation of a comprehensive set of customer-ready documents that describes the implemented solution.

Preproduction Environment

To test and plan the installation of patches and customizations to the different Portal Server components, you must create a preproduction, or staging environment. This environment must be identical to the production environment. Many customers fail to realize that this system is critical to achieve the levels of availability, security, and supportability that they require. Before a patch or fix can be put into the production system, the proper installation process needs to be defined, tested, and documented. This build document should also include how to roll back changes if necessary, and what the expected downtime of the system is. After the patch is installed on the staging server, it must be tested to ensure that there are no new security holes associated with this change.

Look-and-Feel Templates

You should create organization-specific directories that contain the HTML and JavaServer Pages™ (JSP™) technology templates that are used to implement the look and feel of the organization. This is a good practice, even for a server with a single organization.

In the Sun ONE Portal Server 6.2 software, the default desktop templates and JSP technology templates are located in the `/etc/opt/SUNWps/desktop/default` directory. You can copy the contents of this directory to a new directory under `/etc/opt/SUNWps/desktop` and modify the files that are used to implement the desired look and feel. To have the Portal Server to use this directory instead of the default directory, change the desktop type attribute in the organization's Desktop service by using the administration console.

In addition, you should remember that the look-and-feel templates that are included with the Sun ONE Portal Server software are just samples of what can be done with the Sun ONE Portal Server framework. You do not have to use them. In some cases, it will be more efficient to create a complete new and simple set of templates. If this is the case, there is an option during the installation to not install the sample portal.

Even if the sample portal is used, the JSP technology templates need to be cleaned up to remove unused elements. The sample desktop includes several channels and containers just to show what is possible to do. Most of the time, a small fraction of the elements defined in the sample portal are actually used. Streamlining the desktop templates will make the desktop more manageable and easier to maintain. This is also valid for the display profile, which is used to manage the user's visible containers and channels. The default display profile, which is installed with the sample portal, should be trimmed to contain just the necessary definitions.

The HTML and JSP technology templates used during the authentication process are stored in the `/opt/SUNWam/web-apps/services/config/auth` directory. By creating a subdirectory underneath this directory with the modified copies of these templates, you can create a specific look and feel for each organization.

Installation of Customizations

To apply a new set of customizations or patches to a system in production, it is necessary to create a well-defined process to install these customizations on every production server. Often these processes are defined in the Run Books for the system. The processes should define what to do in case the customizations need to be backed out.

It is a good practice to automate the installation process as much as possible through the use of custom installation scripts and to avoid manual steps in the installation process as much as possible. In addition, the installation process should include very detailed installation instructions.

Although they are more complex to create, the best approach is to create custom Solaris OS packages that can be applied, removed, and patched using the standard Solaris OS tools, such as `pkgadd`(1M) and `pkgrm`(1M). This also enables you to quickly verify what version of the customizations are installed by querying the Solaris OS package database.

It is also important to maintain a good tracking system to keep track of changes on the customizations. This enables you to easily roll back changes if regression problems are found. The preferred tools for change control systems and to build applications are the open source tools CVS and ANT.

Documentation

One of the most important mechanisms to increase the supportability of a system is to create a comprehensive set of customer-ready documents that describe the implemented solution. Unfortunately, because the documentation has to be done at the end of the project when time and money are usually running low, the documentation phase is usually reduced to the absolute minimum. The areas that must be documented to create a supportable solution are:

- System architecture
- Software installation and configuration
- Operational procedures (also known as Run Books)
- Software customizations
- Custom code
- Third-party product integration

Third-Party URLs

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

About the Author

Christian Candia has been with Sun Microsystems for more than eleven years in different roles in Engineering, Professional Services, and Sales and Support organizations. Over the last five years, he has focused on customer implementations of the Sun ONE software stack, with special dedication to the Sun ONE Portal Server software and related components. Currently, Christian works as a Solutions Architect for the Sun Professional Services Vertical Solutions Expertise Center where he focuses on creating reference architectures, implementation guides, and best practices to support Sun ONE Portal Server software solutions.

Ordering Sun Documents

The SunDocs^{SM} program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine web site at: `http://www.sun.com/blueprints/online.html`